SERVER_SOURCE

readme

```
Connectivity between the various Perl files

(1) ci.pl
  project.pl
    host.pl
  getsessionid.pl
    project.pl
    webdate.pl
  -> acceptdecline.pl (by way of form submission)

acceptdecline.pl
  cgi.pl
  getsessionid.pl
  project.pl ...
  -> (2a) declined.html OR
  -> (2b) ci.pl

(3) save.pl
  getsessionid.pl

(4) calculate.pl
  project.pl ...
  getsessionid.pl ...
  -> (4a) overuse.html OR
  -> (4b) getresults.pl (by way of META refresh)

(5) getresults.pl
  getsessionid.pl ...
  calculate2.pl
    project.pl ...
    getsessionid.pl ...
    readstuff.pl
      project.pl (?!)
      getsessionid.pl (?!)
    ansoft_files.pl
    remoteTLINE.pl
      getheadpost.pl
```

all_md5

```
MD5 checksums
c4616dce3206bfba1e1d26e910fcda19  readme
f627deb66ff95612e34bbc2840b44069  all_md5
2f73d693d13e2de292c2a9a17b09bb40  acceptdecline.pl
d4d66bd78c7e3d244a844dc8c6568d83  calculate.pl
4f0ef466c56dbf48f431045b3c6b65af  calculate2.pl
a23e93229c5cd96f44ed6bfa8cb43923  cgi.pl
3dd2bd09c6af821ac9489471e0140b51  ci.pl
5f6136da8fbc267052ed9a6e1fcd86c4  getresults.pl
7889f9cab3787db2b0a4231ec128a331  getsessionid.pl
a926a04944ceeeecaf8c5ba09675aa46  host.pl
c04024bba78bf2e5da1fc944f3c083e2  makeauth.pl
ccf4d0e44f6b7b5fa920e23ae022d46b  project.pl
f62c02714876d4686de1139739abccdf  save.pl
d30a1ca9731e7feccf5922c1681192dc  webdate.pl
c2898185d40e61d3e8f98add70d26f4a  native_CrossSection.html
2daeb6d25d7d4ffe52ed835721512f56  nav3_CrossSection.html
cf20dc1f37dc1c1012fdf7fccb8117d1  ../../html/Stackup/declined.html
```

```
90d33ac8912a6b25c3332b34dec14e71  ../../html/Stackup/error.html
cd1410f9391b66a64abe3f0368f72e66  ../../html/Stackup/overuse.html
```

acceptdecline.pl

```
#!/usr/bin/perl

use strict;

my $debug;
#$debug = 1;
if ($debug) { print "Content-Type: text/html\n\n<HTML><BODY><PRE>"; }

require("cgi.pl");

# GLOBALS
use vars qw( $debug );

require("getsessionid.pl");
use vars qw ( $suid $user_id $usage $path_info $save_copy $cookie);

my %values = &process_form();
my $time = hex( &obscure( substr($values{"token"},0,8),
                          substr($values{"token"},8,8) ) );
my $delay = $time+15-time();
if ($debug) { print "time: $time delay: $delay\n"; }

require("project.pl");

use vars qw($HOST_IP);

use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH
$NUM_RECORDS);

if ($values{"choice"} =~ /ACCEPT/) {

#    if ($delay > 0) { sleep($delay); }

    print "Location: http://$HOST_IP/cgi-bin/$PROJECT/ci.pl\n";
    print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
    if ($delay > -300) { print $cookie; }
    print "\n";
}
else {
    print "Location: http://$HOST_IP/$PROJECT/declined.html\n";
    print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
    print "\n";
}
```

calculate.pl

```
#!/usr/bin/perl
# calculate.pl
# Copyright (C) 1999-2000 Rode Consulting, Inc.
# All rights reserved.

use strict;

# GLOBALS
use vars qw( $debug );
#$debug = 1;
```

```
require("project.pl");

use vars qw($HOST_IP);
use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH $NUM_RECORDS);

require("getsessionid.pl");
use vars qw ( $suid $new_suid $user_id $timestamp $usage $valid $path_info $cookie);

$USAGE_MAX=10;
if ($usage >= $USAGE_MAX) {
    print "Location: http://$HOST_IP/$PROJECT/overuse.html\n";
    print "\n";
    exit(0);
}

#print "Location: http://",$HOST_IP,"/cgi-bin/$PROJECT/calculate2.pl\n";
#print "\n";
#
# Rather than directly forward to the script that does the real calculation
# We instead return a small HTML page that displays a message, before
# forwarding to the final location.

my $hash = &hash($user_id);

# Write new cookie to cookie cache
sysopen(AUDIT_DB, "auth.dat", 2) or die("Can't open authorization database!");
sysseek( AUDIT_DB, $hash*$RECORD_LENGTH, 0 );
my $written = syswrite( AUDIT_DB, ($new_suid."\n"),
                        $RECORD_LENGTH );

die "System write error: $!\n" unless defined $written;
close(AUDIT_DB);

print "Pragma: no-cache\nExpires: Mon, 01-Jul-1996 00:00:00 GMT\n";
print $cookie;

print "Content-Type: text/html\n\n";
print <<"EOF"
<HTML><HEAD><TITLE>Extraction in progress...</TITLE>
<META HTTP-EQUIV="refresh" CONTENT="0;URL=http://$HOST_IP/cgi-bin/$PROJECT/getresults.pl">

<SCRIPT LANGUAGE="JavaScript"><!--
        badbrowser = ((navigator.appName == "Microsoft Internet Explorer") && (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") > 0));
        if (badbrowser) this.focus();
        if (!badbrowser && window.focus) this.focus();
//--></SCRIPT>

</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA" ALINK="#FF0000">
```

```
<TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER ALIGN=CENTER>
<CENTER><H2>Please wait...<BR>Extraction in progress</H2></CENTER>
</TD></TR></TABLE>
</BODY>
</HTML>

EOF
    ;
    1;
```

calculate2.pl

```
#!/usr/bin/perl
# calculate2.pl
# Copyright (C) 1998-2000 Rode Consulting, Inc. (RCI)
# All rights reserved.

use strict;

# GLOBALS
use vars qw( $debug );
#$debug = 1;

if ($debug) {
    print "Content-Type: text/html\n";
    print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
    print "\n";

    print "<BODY><PRE>\n";
}

require("project.pl");

use vars qw($HOST_IP);
#use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX $HOST_CGI_PREFIX);
#use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
#use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
#use vars qw(%HTML_MACROS);

use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL);

require("getsessionid.pl");
use vars qw ( $suid $user_id $timestamp $usage $valid $path_info $cookie);

use vars qw( @items $user_id $root $dont_do_it $results);
use vars qw( $units_name $units $scale );

my $LOCK_SH=1; my $LOCK_EX=2; my $LOCK_NB=4; my $LOCK_UN=8;

if (!$dont_do_it && ($usage > $USAGE_MAX)) {
    print "Location: http://$HOST_IP/$PROJECT/overuse.html\n";
    print "\n";
    exit(0);
}

if (!$dont_do_it && ((time()-$timestamp) > 15)) {
    print "Location: http://$HOST_IP/$PROJECT/error.html\n";
    print "\n";
    exit(0);
}
# Retrieve the binary data from a file
    if ( $dont_do_it ||
        ($valid>0 && -e "tmp/stuff".$user_id.sprintf("%04d",$usage)) ) {
#     require("readstuff.pl");
#     $root = "x.pjt";
#     require("field_extractor_files.pl");
# Early return for use with ez2dzip.pl
#     if ($dont_do_it) { return 1; }
#     open(LOCKFILE,">>tmp/lockfile");
#     flock(LOCKFILE, $LOCK_EX);
#     seek(LOCKFILE, 0, 2);
#     require("remoteTLINE.pl");
#     flock(LOCKFILE, $LOCK_UN);
#     close(LOCKFILE);
        open(SAVE,">tmp/results".$user_id.sprintf("%04d",$usage));
        print SAVE "DUMMY";
        close SAVE;
    }
    else {
      print "Location: http://$HOST_IP/$PROJECT/error.html\n\n";
      exit 0;
    }
1;
```

cgi.pl

```
#!/usr/local/bin/perl
#Copyright (C) 1996-8 Rode Consulting, Inc.  All rights reserved.
#package myCGI;
use strict;
# GLOBALS:
# $cgi_query:           The query string (source depends on
REQUEST_METHOD)
#                       (logged by other scripts)
# TEMP %form_values is global for compatibility
# Special (evil?) mode flags:
# $debug:         Global cross-module debug flag
# $_cgi_lowercase:      Form value names are all lowercased
use vars qw($debug $_cgi_lowercase $cgi_query %form_values);
if ($debug) {     print "cgi.pl called\n";            }
# process_form()
sub process_form {
    my ($valid,@pairs) = ("",(""));
```

```
        $| = 1;

    if ($debug) {     print "Method: ", $ENV{'REQUEST_METHOD'}, "<BR>\n";
  }

        $valid = "";
        if ($ENV{'REQUEST_METHOD'} eq 'GET')
        {
          $cgi_query = $ENV{'QUERY_STRING'};
          $valid = ($ENV{'QUERY_STRING'}) ? "true" : "";
        }
        elsif ($ENV{'REQUEST_METHOD'} eq 'POST')
        {
          $valid = ($ENV{'CONTENT_LENGTH'} > 0) ? "true" : "";
          if ($valid) { read(STDIN, $cgi_query, $ENV{'CONTENT_LENGTH'}); }
          $cgi_query =~ s/\n/\&/g;
        }
        elsif ($debug) { print "Unrecognized method:
$ENV{'REQUEST_METHOD'}\n"; }

        if ($debug)   {     print "CGI arguments: ", $cgi_query, "<BR>\n";
  }

        if ($valid)
        {
          @pairs = split(/&/, $cgi_query);
          foreach my $pair (@pairs)
          {
              $pair =~ s/\+\+/\+/g;
              $pair =~ s/\+/ /g;
              my ($name, $value) = split(/=/, $pair);
              $name =~  s/%([a-fA-F0-9]{2})/pack('c',hex($1))/eg;
              $value =~ s/%([a-fA-F0-9]{2})/pack('c',hex($1))/eg;
    $value =~ s/%2[Ff]/\//g;
              if ($_cgi_lowercase) { $name =~ tr/A-Z/a-z/; }
              if ($form_values{$name})
              { $form_values{$name} .= "&" . $value;      }
              else
              { $form_values{$name} = $value; }

              if ($debug)   {     print $name, " = '", $form_values{$name},
"'\n";      }
          }
        }
        %form_values;
    }

    # get_cookie( -cookiename- )
    # Returns the first cookie by the given name

    sub get_cookie {

        my ($cookie) = @_;
        my (@cookies, $cookieName, $cookieVal) = ((""),"","");

        if ($debug) { print "HTTP_COOKIE: ",$ENV{"HTTP_COOKIE"},"\n"; }
        if ($ENV{"HTTP_COOKIE"} =~ /$cookie/)
        {
          @cookies = split(/;\s*/,$ENV{"HTTP_COOKIE"});
```

```
      if ($debug) { print '@cookies: ',@cookies,"\n"; }

      foreach (@cookies)
      {
          if ($_ =~ /$cookie/)
          {
            ($cookieName, $cookieVal) = split (/=/,$_);
            return($cookieVal);
          }
      }
    }
    else { return(''); }
}
##@LOOKUP = split(/Name:/,'nslookup $ENV{"REMOTE_ADDR"}');
##@LOOKUP = split(/Address:/, $LOOKUP[1]);
##$client_name = $LOOKUP[0];
#$client_name = $ENV{"REMOTE_ADDR"};
#$client_name =~ s/ //g;
#package form;
1;
```

ci.pl

```
#!/usr/bin/perl

use strict;

# GLOBALS
use vars qw( $debug );

my $debug;
#$debug = 1;
if ($debug) { print "Content-Type: text/html\n\n<HTML><BODY><PRE>"; }

require("project.pl");
use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX $HOST_CGI_PREFIX);

use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH $NUM_RECORDS);

require("getsessionid.pl");
use vars qw ( $suid $user_id $usage $path_info $save_copy $cookie);

if ($debug) {
    print $suid," ",$user_id," ",$usage," ",$path_info," ",$cookie;
}

print "Content-Type: text/html\n";
print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
print "\n";

if (length($suid) > 0) {

    if ( !($ENV{"HTTP_USER_AGENT"} =~ /3.0/) ) {
      open(HTML, "native_CrossSection.html");
    }
    else {   open(HTML, "nav3_CrossSection.html");   }
    read(HTML, my $html, 1000000);
    close(HTML);

    $html =~ s/<HOST_IP>/$HOST_IP/g;

    print $html;
}
else { # New cookie

      open(DISCLAIMER, "../../html/netsim/termsofuse.html");
      read(DISCLAIMER, my $disclaimer, 1000000);
      close(DISCLAIMER);

      my $time = sprintf("%08x",time());
      my $mask = substr(&sumcheck($time),0,8);
      my $token = &obscure($time,$mask).$mask;
      $disclaimer =~ s/<!-- marker1 -->/  \(Please read carefully then
click <B>ACCEPT<\/B> or <B>DECLINE<\/B> at the bottom of this page\)/;

      my $replacement = <<"EOF"

<CENTER>
<APPLET CODE="AcceptDecline.class" CODEBASE="http://$HOST_IP/$PROJECT/"
WIDTH=250 HEIGHT=40>
<PARAM NAME=token VALUE="$token">
</APPLET>
</CENTER>
EOF
      ;
      $disclaimer =~ s/<!-- marker2 -->/$replacement/;

      print $disclaimer;
}

exit(0);
```

getresults.pl

```
#!/usr/bin/perl

use strict;

require("getsessionid.pl");
use vars qw( $debug );
use vars qw ( $suid $new_suid $user_id $usage $path_info $save_copy
$cookie);

my $results_filename = "tmp/results".$user_id.sprintf("%04d",$usage);

if (! (-e $results_filename) ) {
    open(SAVE,">$results_filename");
    close(SAVE);
    require "calculate2.pl";
}

print "Content-Type: text/html\n";
print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
print "\n";
print "<HTML><HEAD>\n";
print "<TITLE>Field Extraction Results</TITLE>\n";
```

```
        print "<SCRIPT LANGUAGE=\"JavaScript\"><!--\n";
        print "     badbrowser = ((navigator.appName == \"Microsoft Internet
Explorer\") && (parseInt(navigator.appVersion) == 4) &&
(navigator.appVersion.indexOf(\"Mac\") > 0));\n";
        print "     if (badbrowser) this.focus();\n";
        print "     if (!badbrowser && window.focus) this.focus();\n";
        print "//--></SCRIPT>\n";
        print "<STYLE TYPE=\"text/css\">\nBODY    { font-size:10pt }\n    H1
   { font-size:24pt }\n    H2    { font-size:20pt }\n    H3    { font-size:16pt
}\n   TABLE { font-size:10pt }\n    TD    { font-size:10pt }\n</STYLE>\n";
        print '</HEAD>';
        print '<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF"
VLINK="#AA00AA" ALINK="#FF0000">',"\n";

        my $results;
        open(STUFF,$results_filename);
        read(STUFF,$results,1000000);
        close(STUFF);

        if ($results =~ /ERROR/ || length($results) == 0) {
            print "<CENTER>\n";
            print "<TABLE><TR><TD WIDTH=\"80%\">\n";
            print "<BR><H3>We're sorry, but an error occurred attempting to
extract this geometry.<BR>  Please check for overlapping elements and try
again.</H3>\n";
            print "</TD></TR></TABLE>\n";
            print "</CENTER>\n";
            print "</BODY></HTML>\n";
            exit 1;
        }
        else {

        # POST PROCESS RESULTS INTO HTML (THIS IS A PLACEHOLDER)
            print "<CENTER>\n";
            print "<H3>Complete</H3>";
            print "</CENTER>\n";
            print "</BODY></HTML>\n";
            exit 0;
        }

        sub pretty() {
            my $precision = pop(@_);
            my $number = pop(@_);
            my $index = index $number,"e",0;
            if ($index < 0) { $index = index $number,"E",0; }
            if ($index>0 && $index > $precision) {
              return substr($number,0,$precision).substr($number,$index);
            }
            else { if ($index>0) { return $number; }
                 else { return substr($number,0,$precision); }
              }
        }
```

getsessionid.pl

```
        #!/usr/bin/perl
        #Copyright (C) 1997-2000 Rode Consulting, Inc.
        #All rights reserved.

        use strict;
```

```
use MD5;

require("project.pl");

use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
$HOST_CGI_PREFIX);
use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
use vars qw(%HTML_MACROS);

use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $NUM_RECORDS
$RECORD_LENGTH);

require("webdate.pl");

# GLOBALS
use vars qw( $debug );
use vars qw ( $suid $new_suid $user_id $timestamp $usagestamp $usage
$valid $path_info $cookie);

my $COOKIE_LENGTH = 35;

#$debug = 1;
if ($debug) {
    print "Content-type: text/html\n\n<HTML><BODY><PRE>\n";
    print "path_info: $ENV{'PATH_INFO'}\ncookie: $ENV{'HTTP_COOKIE'}\n";
}

# Any old random number will do (32 bytes)...
my $local_cryptkey =
"baceeed5e3ff05b81b3688c1e10b914bd2a1518edb6c9090358eb21ccef6da82";

sub obscure() {
    my $mask = pop @_;
    my $target = pop @_;

    my $obscured = "";
    for (my $i=0; $i<length($target); $i++) {
      $obscured .= sprintf( "%1x", hex(substr($target,$i,1)) ^
hex(substr($mask,$i,1)) ^ hex(substr($local_cryptkey,$i,1)) );
    }

    return $obscured;
}

sub sumcheck() {
    my $source = pop @_;

#    my $sum = ‘echo $source | md5sum 2>/dev/null‘;
#    $sum =~ s/\s*-\s*//g;
    my $md5 = new MD5;
    $md5->add($source."\n");
    my $sum = unpack( "H32", $md5->digest() );

    return $sum;
}

sub encode_suid() {
    my $usage = pop @_;
```

```
                my $usagestamp = pop @_;
                my $timestamp = pop @_;
                my $user_id = pop @_;

            #open(LOGFILE,">>logfile");
            #print LOGFILE "E $user_id $timestamp $usagestamp $usage\n";
            #close(LOGFILE);

                my $combined =
$user_id.sprintf("%08x%08x%04d",$timestamp,$usagestamp,$usage);
                my $sum = &sumcheck($combined);

                $sum =~ s/^0*//g;
                $sum = substr($sum.$sum,0,$COOKIE_LENGTH);

                my $obscured = &obscure($combined,$sum);
                return $obscured.$sum;
            }

            sub decode_suid() {
                my $encoded = pop @_;

                my $unobscured =
&obscure(substr($encoded,0,$COOKIE_LENGTH),substr($encoded,$COOKIE_LENGTH,$COOKIE_LE
NGTH));

                my $user_id = substr($unobscured,0,15);
                my $timestamp = hex substr($unobscured,15,8);
                my $usagestamp = hex substr($unobscured,23,8);
                my $usage = hex substr($unobscured,31,4);

            #open(LOGFILE,">>logfile");
            #print LOGFILE "D $user_id $timestamp $usagestamp $usage\n";
            #close(LOGFILE);

                my $valid = 0;
                if ( $encoded eq &encode_suid($user_id, $timestamp, $usagestamp,
$usage) ) { $valid = 1; }

                return ($user_id, $timestamp, $usagestamp, $usage, $valid);
            }

            # GET USER_ID, USAGE AND PATH_INFO

            $path_info = $ENV{'PATH_INFO'};
            $path_info =~ s/[<>!&\~]//g;
            $path_info =~ s/\.{2,}//g;    # don't allow uptree accesses
            $path_info =~ s/\/{2,}/\//g;
            $path_info =~ /()/;  # Clear $1
            $path_info =~ s#/([^\/]*)/?##;

            $valid = 0;
            $suid = "";
            if ( $ENV{'HTTP_COOKIE'} =~ /SUID2\s*=\s*(\w*)/ ) {
                $suid = $1;
                $suid =~ s/\s*//g;

                ($user_id, $timestamp, $usagestamp, $usage, $valid) =
&decode_suid($suid);
```

```
      # Some of the cookies are cached in a database.
      # This code is intended to randomly audit the cookie data
      # to make sure it isn't corrupt or counterfeited.

        sysopen(AUDIT_DB, "auth.dat", 2) or die("Can't open authorization
database!");
        my $hash = &hash($user_id);
        sysseek( AUDIT_DB, $hash*$RECORD_LENGTH, 0 );
        sysread( AUDIT_DB, my $record, $RECORD_LENGTH-1 ); # skip newline
        (my $prev_user_id, my $prev_timestamp, my $prev_usagestamp, my
$prev_usage, my $prev_valid) =
          &decode_suid($record);
        close(AUDIT_DB);

        if ( $prev_valid && ($prev_user_id eq $user_id) &&
          ($prev_usagestamp eq $usagestamp) && ($usage < $prev_usage) ) {
          $valid = 0; # Apparently the cookie is fake
        }

      # If more than 24 hours (or some other time limit) has passed,
      # reset the usage count
        if ( (time()-$usagestamp) > $USAGE_INTERVAL ) {
          $usagestamp = time();
          $usage = -1;
        }

        if ($usage > $USAGE_MAX) {
          $valid = 0;
        }

      # DOS inhibition
        if ( !( $ENV{"HTTP_USER_AGENT"} =~ /Mozilla/ ) ) {
sleep(15+rand(10)); return 0;  }
      }
      else { # No cookie - create a new one
        (my $ip = $ENV{"REMOTE_ADDR"}) =~ s/\./:/g;
        (my $a, my $b, my $c, my $d) = split(":",$ip);
        $ip = ((($a*256+$b)*256+$c)*256+$d);
        my $rand = int(rand(2**24));
        $user_id = sprintf("Z%08x%06x",$ip,$rand);

        $usagestamp = time();
        $usage = -1;
      }

      $new_suid = &encode_suid($user_id, time(), $usagestamp, $usage+1);

      # Expire cookie in 90 days
      $cookie = "Set-Cookie: SUID2=$new_suid;
expires=".&RFC950date(time()+7*24*3600)."; path=/cgi-bin/$PROJECT/;
domain=$HOST_IP\n";

      if ($debug) { print "Using PATH_INFO...\n"; }

      if ($debug) { print "SUID: ",$suid," ","PATH_INFO: ",$path_info,"\n"; }
      1;
```

host.pl

```
#!/usr/bin/perl
#Copyright (C) 1997-2000 Rode Consulting, Inc.
#All rights reserved

use strict;

use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX $HOST_CGI_PREFIX);
use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
use vars qw($BODY $NOCACHE_HEADER);
use vars qw(%HTML_MACROS);

$DOMAIN_NAME="circuitsim.com";
$HOST_NAME = "www.".$DOMAIN_NAME;
$HOST_IP = "24.218.251.135";

my $PREFIX = "Stackup/patent";
$HOST_PREFIX = "http://$HOST_NAME/$PREFIX/";
$HOST_CGI_PREFIX = "http://$HOST_NAME/cgi-bin/$PREFIX/";
$HOST_PREFIX_JAVA = "http://$HOST_IP/$PREFIX/";
$HOST_CGI_PREFIX_JAVA = "http://$HOST_IP/cgi-bin/$PREFIX/";
$BODY = '<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA" ALINK="#FF0000">';
$NOCACHE_HEADER = "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";

$HTML_MACROS{"DOMAIN_NAME"} = '$DOMAIN_NAME';
$HTML_MACROS{"HOST_NAME"} =   '$HOST_NAME';
$HTML_MACROS{"HOST_IP"} =     '$HOST_IP';
$HTML_MACROS{"HOST_PREFIX"} =       '$HOST_PREFIX';
$HTML_MACROS{"HOST_CGI_PREFIX"} =   '$HOST_CGI_PREFIX';
$HTML_MACROS{"HOST_PREFIX_JAVA"} =  '$HOST_PREFIX_JAVA';
$HTML_MACROS{"HOST_CGI_PREFIX_JAVA"} =    '$HOST_CGI_PREFIX_JAVA';
$HTML_MACROS{"BODY"} =        '$BODY';
$HTML_MACROS{"TIME"} =        '$TIME';
$HTML_MACROS{"SESSIONID"} =   '$session_id';
$HTML_MACROS{"SESSIONIDMACRO"} = '$session_id_macro';
1;
```

makeauth.pl

```
#!/usr/bin/perl

require("project.pl");

my $nothing = "";
for (my $i=0; $i<$RECORD_LENGTH-1; $i++) {
    $nothing .= " ";
}

open(AUTH, ">auth.dat");

for (my $i=0; $i<$NUM_RECORDS; $i++) {
    print AUTH $nothing,"\n";
}
```

project.pl

```
#!/usr/bin/perl
#Copyright (C) 1997 Rode Consulting, Inc.
#All rights reserved.
```

```
use strict;

use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $NUM_RECORDS $RECORD_LENGTH);

$PROJECT = "Stackup";
$USAGE_MAX = 10;
$USAGE_INTERVAL = 12*3600;
$NUM_RECORDS = 10000;
$RECORD_LENGTH = 71;

use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX $HOST_CGI_PREFIX);
use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
use vars qw(%HTML_MACROS);

# Not a general purpose hash -
# intended to use positions 5-11 inclusive of user_id
sub hash() {
    my $seed = pop @_;
    srand( hex substr($seed,5,7) );
    return int(rand($NUM_RECORDS));
}

require("host.pl");

1;
```

save.pl

```
#!/usr/bin/perl
# Copyright 1999-2000 Rode Consulting, Inc.
# All rights reserved

use strict;

require("getsessionid.pl");
use vars qw( $debug );
use vars qw ( $suid $user_id $timestamp $usage $valid $path_info $cookie);

my $cgi_query;
read(STDIN, $cgi_query, $ENV{'CONTENT_LENGTH'});

print "Content-Type: text/plain\n";
print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
print "\n";

if (length($user_id) > 0) {
    ‘rm -f tmp/*$user_id*‘;
    open(STUFF,">tmp/stuff".$user_id.sprintf("%04d",$usage+1));
    print STUFF $cgi_query;
    close(STUFF);
    print "OK\n";
}
else { print "ERROR\n"; }
```

webdate.pl

```
#!/usr/bin/perl
#Copyright (C) 1997 Rode Consulting, Inc.
#All rights reserved.

use strict;

# GLOBALS
use vars qw( $debug );

sub zeropad2 {
    my $number = pop(@_);
    my $padded;
    if ($number < 10) { $padded = "0".$number; }
    else { $padded = "" . $number; }

    $padded;
}

sub RFC950date {
    my $time = pop(@_);
    (my $sec,my $min,my $hour,my $mday,my $mon,my $year,my $wday,my $yday,my $isdat) = gmtime($time);
    if ($year < 1970) { $year += 1900; }
    if ($year < 1970) { $year += 100; }

#     $formatted='/usr/local/bin/date -d '$time seconds' '+%A, %d-%b-%y %T GMT' --universal';
    ("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")[$wday]."," ".
        &zeropad2($mday) . "-" .
    ("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")[$mon] . "-" .
        $year . " " .
        &zeropad2($hour) . ":" . &zeropad2($min) . ":" . &zeropad2($sec) . " GMT";
}

sub modified {
    my $file = pop(@_);
    (my $dev,my $ino,my $mode,my $nlink,my $uid,my $gid,my $rdev,my $size,my $atime,my $mtime,my $ctime,my $blksize,my $blocks) = stat($file);

    if ($debug) { print "Modified: ",$mtime," ",time(),"\n"; print $^T; }

    &RFC950date($mtime);
}

1;
```

native_CrossSection.html

```
<HTML>
<HEAD>
<TITLE>EXPERIMENTAL PCB Cross Section Editor</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA" ALINK="#FF0000">
```

```
<APPLET CODE="CrossSection.class" CODEBASE="http://<HOST_IP>/Stackup/"
ARCHIVE="CrossSection.jar" WIDTH=1024 HEIGHT=540>
<PARAM NAME=addgeometry VALUE=ENABLE>
<PARAM NAME=userid VALUE="00000000">
</APPLET><BR><BR>

<!-- Instructions deleted -->

</BODY>
</HTML>
```

nav3_CrossSection.html

```
<HTML>
<HEAD>
<TITLE>EXPERIMENTAL PCB Cross Section Editor</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

<APPLET CODE="CrossSection.class" CODEBASE="http://<HOST_IP>/Stackup/"
ARCHIVE="CrossSection.zip" WIDTH=1024 HEIGHT=540>
<PARAM NAME=addgeometry VALUE=ENABLE>
<PARAM NAME=userid VALUE="00000000">
</APPLET><BR><BR>

<!-- Instructions deleted -->

</BODY>
</HTML>
```

../../html/Stackup/declined.html

```
<HTML>
<HEAD><TITLE>Declined</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

Sorry you feel that way...

</BODY>
</HTML>
```

../../html/Stackup/error.html

```
<HTML>
<HEAD>
<TITLE>Extracting geometry...</TITLE>
<SCRIPT LANGUAGE="JavaScript"><!--
      badbrowser = ((navigator.appName == "Microsoft Internet Explorer")
&& (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") >
0));
      if (badbrowser) this.focus();
      if (!badbrowser && window.focus) this.focus();
//--></SCRIPT>
</HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">
<TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER
ALIGN=CENTER><CENTER>
<H2>We're sorry an ERROR occurred in processing your data.<BR>
Please try again later.</H2>
```

```
<H3><I>Cookies must be enabled!</I></H3>
</CENTER></TD></TR></TABLE>
</BODY>

</HTML>
```

../../html/Stackup/overuse.html

```
<HTML>
<HEAD>
<TITLE>Extracting geometry...</TITLE>
<SCRIPT LANGUAGE="JavaScript"><!--
      badbrowser = ((navigator.appName == "Microsoft Internet Explorer") && (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") > 0));
      if (badbrowser) this.focus();
      if (!badbrowser && window.focus) this.focus();
//--></SCRIPT>
</HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA" ALINK="#FF0000">
<TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER ALIGN=CENTER><CENTER>
<H2>We're sorry but you've exceeded your<BR>extraction quota for today.<BR><BR>
Please try again tomorrow.</H2>
</CENTER></TD></TR></TABLE>
</BODY>

</HTML>
```